

Ingeniería y Calidad de Software

15/08

Mundar Mail a las 3 a la vez
imelis@gmail.com
ceciliamassano@gmail.com

Foco de la materia en el proyecto

14 de Noviembre Última actividad con calificación

AD 8 o más en cada parcial \rightarrow 1 recuperatorio y queda la mejor
TP prácticas grupales con más de 8
TP de investigación grupo nota 8 superior \rightarrow 1 recuperatorio
Regular Aprobados los 2 parciales \rightarrow 1 recuperatorio
TP grupales aprobados (prácticas y de investigación)

Grupos de 5 o 6

13 ejercicios - Algunas son prácticas
9 evaluables en grupo, presentar si o si todas
Hay que aprobar el 70% de esos 9.

Guía de ejercicios resueltos para practicar

Parciales son los sábados

Se aprobó con 4 en parciales (SS 7%)

Primeros 23 de Septiembre } A las 9 hs
Seg 28 de Octubre
Tercer 11 de Noviembre }

TP de Investigación

- 1 - Viernes 3 de Noviembre (Poster Testing Agil) \rightarrow poster científico (subir PDF onto de la expo)
- 2 - Viernes 10 de Noviembre (charla TED)

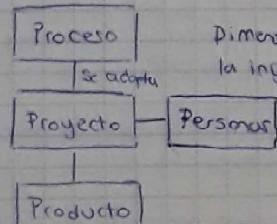
Examen Final

Tema sale sorteado un mes antes (o cuando te inscribas)

Después preguntan temas de cualquier unidad

Puede hablar uno o varios

User stories
Filosofías Lean y agil
Apuntamos ahora a los roles de coordinación



Dimensiones sobre las que trabaja la ingeniería de software

: Volar cosas al repositorio \rightarrow empezar con el libro.

Aseguramiento de calidad se refiere a si hizo las cosas como se esperaba
Los productos deben cubrir los requerimientos (no es asegurar la calidad del producto).

Guía de ejercicios evaluables → NO se suben soluciones

3 disciplinas sobre las que func. la Ing en soft. (3P)

Proceso: la más abstracta y producto, el más concreto.

→ (planeo, organización, seguimiento)

Proyecto: es una unidad de gestión y organización de recursos y personas para obtener como resultado un producto o servicio.

Rol de Ing de procesos:

Rol de líderes de proyecto

Procesos

Definidos PUD

RUP

Definición de un proceso:

Intención de considerar todos los factores necesarios para hacer soft (artefactos, salida, roles, métricas)

→ basado en experiencia

Procesos Empíricos

Empoderar a los equipos de proyecto

Ágil

sin pensamiento

o filosofías, no

método/lenguaje

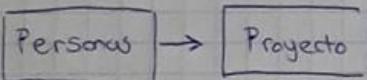
Procesos definidos
buscan repetitividad para inferir.

→ Corriente
único estm
mejoradas

Personas son el elemento fundamental para hacer software

(el mayor costo (80% esfuerzo))

hs persona lineales



Problemas de comunicación para empezar en la toma de requerimientos

Procesos empíricos: se basan en 3 pilares

Inspección

Adaptación

Transparencia

como neces. aprendes de
la experiencia neces exp
rápido. ciclos cortos.

Se debe trabajar con ciclos iterativos si o si (no puede ser cascada)

No importa cuál ciclo iterativo pero debe ser iterativo.

Mientras más corta la iteración mejor. Porque nos dará cuenta antes del error

Manifiesto Ágil → compuesto por principios y valores

manifestoagil.org (lo vamos a tener que leer)

Scrum habla de como gestionar proyecto no de construir productos.

Infografía → se priorizan las tareas.

El equipo de proyecto → compone por gente de negocio y gente técnica

Product Owner: está al lado del negocio: es él que tiene las necesidades, forma parte (esto no se consideraba en los procesos definidos).

→ debe tener capacidad de toma de decisiones para que sea útil.

Aceptar que los req. cambian
Ir y preguntarle al cliente, 50% de los req aparecen cuando el cliente ve el soft.

Empoderamiento de los miembros del equipo
Individuos motivados, capacitados y con autogestión, también la auto-organización

Las mejores ideas salen del equipo
Darse al equipo capa ↑
ciudad de toma de decisiones

El mejor método de comunicación es cara a cara
Importancia de tener equipos co-located (en la misma ubicación física)
Si no puede estar que llegue tras llamarlo a ^{que el} cliente
poco tiempo

Agile

Su manifiesto funciona como un paraguas que abarca a muchos frameworks
nosotros hacer entregas frecuentes de lo que es de valor para el cliente

Requerimientos Agile

El software no es un fin en si mismo es una herramienta para que nuestros clientes
tengan valor de negocio.
Conocer el negocio para poder construir el producto de software correcto.

Mike Cohn → referente de User Stories

Definir lo que es suficiente - Vamos a entregar poco funcionalidad cada vez
45% de las func de un prod de soft no se usa nunca
Hacer foco en entregar MVP (producto mínimo viable)
↓ porcentaje de func que si son necesarias
Trabajar haciendo un producto que no está completamente definido
Product Backlog; lista priorizada de características del producto.
Están ordenadas por lo que el product owner necesita antes que lo otro.
El dueño del product backlog es el product owner, puede hacer cambios cuando
quiero. Los req emergentes van cayendo ahí y se van sucediendo los que ya se trabajaron

Just In Time

No invertir tiempo en expectativas de req que no vamos a necesitar.
Hay que analizarlas cuando se las necesita.

Eliminar desperdicios

Evitar

BRUF invertir tiempo en buenas grandes req cuando el 80% no sirve.

Los requerimientos tienen dist. grados de granularidad.

Users stories → hablamos de ~~reg de usuario~~ reg de usuario. ↗ reg de soft
↓ los debe escribir el product owner (por lo menos la idea debe venir de él, se lo
puede aclarar).

3 variables que considerar y regular:

- Alcance
- Costos
- Tiempo

Con formato: Espacio Despues: 0 pto

definido → dirigido por un plan
Una vez definidos los req no cambian los rec y tiempo se optimizan.

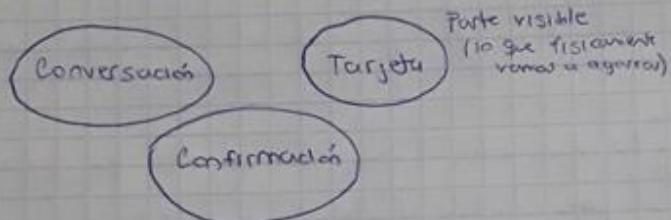
Agil o dirigido por valor:
Define una iteración de tiempo fijo (e.g. un mes) ahí se acuerda juntas que puede entregar. Variables → estar permeables y cambiables.

Principios son los que fundamentan las user stories.

User stories: descripción corta de una func definida para el usuario.

No existen labores de planificación → lo más difícil es decidir qué irá hacer.
(no sirve bullet) → La pregunta es el técnico sobre este artefacto.
Habla de los casos excepcionales y accidentales del soft.

Partes de una user story



Quién, qué, para qué

Es importante respetar la sintaxis de las user stories

Los usan el proyecto a las user stories
Hay algunas cuya granularidad es más gruesa (épicas o temas)
Arrivan en el product backlog siempre va a haber user stories.

Porciones verticales → De incluir un pedacito de cada capa.

Criterios de aceptación

Son límites que pongo sobre los users para que no queden dudas en algunos requerimientos.
Son los que nos ayudan a armar las pruebas de aceptación.

Requerimientos Ágiles

22/08

Cuando veíamos los cu especificábamos los req

En las user stories pienso el como recien cuando lo voy a programar. Más que nada lo que buscamos es identificar
Apuntamos a el punto o el último momento para especificar.

Una user ~~story~~ ^{conversación} story es un token para una ~~función~~. Es una conversación con el usuario.

El equipo define el template de la user story

Frase Verbal

Como [rol] quiero/puedo [acción/que] para
[VIVIR de negocio]

Sirven para definir los
problemas

• cosas que
tienen que
poner para
que el usuario
acepte
(reglas de
negocio especif
vinculadas a
este user y RNF)

Criterios de Aceptación [...]

Pruebas de Aceptación (en como vivir de un)

- Exito
- Falla

Lo debe
aceptar el
Product Owner
(ni cliente,
ni usuario)

El cliente es
U que pone
la pluma

→ Lo importante
es que sepa
del negocio

Los históricos tienen que ser independientes
Mantenerse satisfechos de tener el pro-

yecto de acuerdo al criterio de diseño

Festivales de Folklore

Vender Entradas (que se acepta)

como Vendedor quiero vender las entradas de un festival
para que los clientes puedan ingresar

Criterios de Aceptación

- Venta unica en efectivo
- el número de entradas vendidas debe ser mayor a com
y menor al número de entradas disp

Pruebas de aceptación

• Probar si vende entradas cuando no hay disp (falla)

• ↓ ↓ ↓ ↓ hay disp (pasa)

• ↓ ↓ ↓ ↓ ingresando un monto de plato mayor al total (pa)

• ↓ ↓ ↓ ↓ ↓ ↓ menores ↓ ↓ (falla)

• Probar que se genere la entrada en 6 seg (pasa) (n si podes poner en mas)

Situaciones de falla (situaciones en las el soft hace bien usando debemos fallar),
que no pasa algo que no debería pasar

Aplicación

Mis Gastos

Tareas

• Iniciar sesión

• Registrar gastos

• Administrar tipo de gasto (soporte)

• Registrar Usuario

• Consultar gastos (o visualizar)

• Filtrar gastos

• Cerrar Sesión

• Administrar responsable de gasto

una iteración dura lo que

dura un equipo (entre 1 y 4 semanas)

100% de trabajo

100% compromiso

100% compromiso

Registro Gasto

Como Responsable de gastos quiero registrar un gasto para una actividad realizada fecha para poder gestionar los gastos de mi familia.

Criterios de aceptación:

- La fecha del gasto debe ser anterior o igual a la actual
- Se debe ingresar un monto positivo
- Se puede combinar el responsable
- Se debe especificar el tipo de gasto
- Se debe poder filtrar, ordenar y agrupar mostrando coincidencias

Pruebas de Aceptación:

- Probar ingresar un monto positivo al registrar un gasto (paso)
- Probar ingresar un monto negativo al registrar un gasto (fallo)
- Probar ingresar una fecha posterior de la actual (fallo)
- Probar ingresar una fecha anterior (paso)
- Probar combinar el responsable de gasto (paso)
- Probar ingresar primera 3 letras y ver muestra responsable de gasto (paso)
- Probar ingresar sin monto (fallo)

12/09 Primer ejercicio evaluable
Implementación de User.

25/08

Requerimientos Ágiles

funcionando ya:

Entrega de valor de negocio

funcionando ya:

Entrega frecuente de software funcionando

El producto no está 100% definido

(el soft es un medio para un fin, no es el fin en si mismo)

Difícil que el cliente acepte iniciar mas el proyecto con una visión global y algunas req.

vision
de uti
z. en u

Product Backlog
1. Muestra Tareas
2. 100% defin.
req. no cumplida
(no cumpl.)

Eliminar Desperdicios.

Diferir las decisiones JUST in time

En el contexto de las req. Ágiles surgen las User stories.

(vemos = especificar bien lo necesario lo que necesitamos)

estimar → complejo

↳ universo de incertidumbre

2 principios de Leen que se tomaron en

- Eliminar desperdicios

(se pierden mucho tiempo en espec de retroal. esto orig. de que después cambian de opinión).

- Diferir las decisiones

hasta el último momento respuesta

Postergar hasta tener un poco más de información para la toma de decis.

Leyendo:
Todavía se puede dec. dif.

USER STORY

Card

Conversación
Confirmación

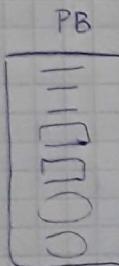
No es un artefacto de SCRUM

La config del product backlog van cambiando, pueden surgir defectos.
Los items del product backlog pueden tener distintas granularidades (US, Epica, Tarea).

SPIKE → tarea de IN

User que tiene tal incertidumbre que no se puede estimar.

Criterio de Lista (Do Ready) : se usa para evaluar la calidad de la historia que escribió. Tiene un criterio de minimo (modelo de INVEST)



Se programan
no designadas.

es un
contenedor
de los req
Tiene
US, Stories, Epicas,
tareas, SPIKES,

tech stories, de
factos, dudas... Si no puedo estimar
nla es porque hay demasiada
se llaman items de incertidumbre

Deben estar priorizadas
y deben estar estimadas

Los FRS intenta
vivir mientras
vive el producto

Los SPIKE son
muy siendo 1
= más US's.

Do Ready

> una vez que se definió están visibles para que sepas que hacer.

Ler de SCRUM (guia oficial)

que el user define sus
jerga, lim. tecnologico.

Modelo INVEST

Independiente (el user pueda ser elegible en cuadq ordenes) → por el product owner cuando pide.

Negociable : Debe expresar una necesidad definiendo el "que" para ofrecer opciones de "cómo".

Valuable : Esto reflejado en el punto, debe entregar valor.

Estimable : Que yo esté en condic de asignarle valor numérico si una user.

Small (Pkg) → La reg a consumir en una iteracion (variable duracion, rues).

Testable : Puedo mostrar que es verificable

Si no puedo estimar
factor, dudas... → Ahí aparece el concepto de SPIKE, lo pude tener en el backlog.

Le mides un tiempo a la iteracion y ves que hacer después

SPIKE → pueden ser de 2 tipos (es un item de investigación)

→ Funcional (def de neg que neces y no teneras) → Gue

→ Técnicas (tecnologias o algoritmo que no estuve pudiendo resuver) → Gue

Para saber si realmente lo terminaste → Do Done (Definición de terminado).

Cuando estimas no solo se debe considerar el de codificación (que sera un 30%), lo otro es pensar, la experiencia de usuario... entre otras...

Deberá haber revisiones de pares antes de subir el código.

Estimaciones ligeras : para alineadas con el pensamiento ágil.
usa técnica Poker Planning o poker estimation

Estimación : predicción de un determin valor. Para saber el como y cuanto necesitas saber otras cosas como el que.

El costo es lo último que se estima.

apartado C) proceso de estimar

Estimaciones Ágiles.

- Relativa (con un marco de referencia, seguimos un item contra el cual comparar).
- Foco en la certeza (No en la precisión porque es corta y si excede se desperdicia) (Nivel de detalle menor ayudando con comparativa y menor posibl. de error).
- ~~Equipo~~ Estima quien hace el trabajo (estima el equipo)
se usan para medir trabajos cortos.

US → Story Point
(Unidad de estimación de la US)
US para estimar → Dependencia del equipo
tamaño)

Ese horario se ve limitado por 3 dimensiones. (se basa en Story Point)
Instalación (mayor homogeneidad)
Complejidad (media)
Esfuerzo (se mide en horas idóneas)- las personas tienen
Duda (incertidumbre),
se compara con la comodidad que se elige primera.

Serie de Fibonacci crecen expon.

Se suman los anteriores para lograr el sig.

Se puede simplificar pero hasta un determinado punto.

Poker Planning

- US
- Serie de Fibonacci: $0, 1, 1, 2, 3, 5, 8, 13, 21 \dots$
 - Wide Band Delphi (como base para turnar la dinámica de estimación)
Alrededor de 2 iteraciones por US

Canónica: Es una usd de este producto no necesaria la primera que se va a implementar. Se compara después contra esta, tiene valor 1 en general pero puede ser de 3. Se define una sola vez.

29/08

Deliveryfat

Roles

Cliente

Comerciante

- Registrar pedido a comercio adherido ✓
Iniciar sesión ✓
Administrar formas de pago X No lo maneja
consultar locales adheridos ✓
consultar seguimiento de pedido ✓
Notificar entrega de pedido (el cliente cuando paga) ✓
Agregar producto a carrito ✓
Administrar locales adheridos ✓
Registrar usuario ✓ 3 usd
Registrar pedido a locales no adheridos ✓
consultar pedidos pendientes ✓
Modificar prod del carrito ✓
Quitar prod de carrito ✓
Consultar pedidos ✓
Recibir notif prep de pedido ✓
Recibir notif finalizado de pedido ✓

3 Agregar producto al carrito → escribir
2 Modificar " "
1 Quitar " " } Estimar

Lógica → Registrar usuario
Con Mail

Agregar producto al carrito]

Como cliente quiero agregar un producto al carrito para realizar mi pedido.

3

Criterios de aceptación:

La cantidad de unidades debe ser mayor a cero y menor que el stock.

Se puede agregar observaciones.

El producto a agregar debe ser del mismo catálogo.

El volumen del producto a agregar sumado con los productos ya seleccionados no debe superar el volumen de la mochila.

Pruebas de aceptación:

Probar agregar un producto al carrito cuya cantidad sea mayor a cero, menor que el stock actual y cuyo volumen entre en la mochila (Pasa).

Probar agregar cero unidades (falla)

Probar agregar un producto cuyo volumen supere el de la mochila (falla)

Probar agregar un producto de otro catálogo (falla)

Probar agregar una cantidad de unidades de producto que supere el stock actual (falla).

- Registrar pedido de lo que sea

1/09

Gestión de productos

Producto mínimo viable (MVP) → definición va al punto práctico → User Stories

U1, punto U2 y U3 en el teórico

Se empezó a fusionar el concepto de producto y de proyecto.

Cuando termina un proyecto termina una versión de un producto y si quiere modificar algo empieza otro proyecto.

No sabemos si vamos a ser quienes mantenga o modifique el soft por esto debemos documentar.

Darle importancia al producto y tener a alguien que vele por su crecimiento.

Surgen roles nuevos para la gestión de productos.

Líder de proy, podría ser un ej de un rol que surge con los proy.

Product Manager → tiene que velar por el crecimiento del producto.

Tendrá que negociar

Product owner: rol técnico necesariamente. Debe venir de las áreas de negocio.
Surge de las startups por ej. esto de enfocarnos en el producto. Toman filosofías
Lean y Ágil para evaluar.

Eric Ries → inventó este concepto MVP → Lean startups.

Distintas formas de establecer matrices de contraste
visibilidad de la realidad → número para que no sea subjetiva
pueden ir las usuarios, la tasa de retención (que no desinstales), de uso

Vamos a entregar el producto por partes → problema de priorización
MVP → ayuda a tomar la decisión.

Pensar para un producto que voy a salir a vender no uno hecho a medida.

los clientes son cada vez menos cautivos.

Visibilidad → definir esp. de usuario

Si vas de abajo hacia arriba es más difícil saltar la brecha → se siente.

Si empiezas desde arriba focalizándote en la experiencia la brecha es más fácil de saltar. Un ej. de un soft signif. es WhatsApp.
se espera una evolución en el software y va a depender de donde empieza.

Queremos crear un prod. único y saber a cuantos usuarios va a llegar, lo van a usar y con qué inversiones? Entonces necesitamos un prod. más chiquito (MVP).
Tenemos la hipótesis y agrupar un grupo de usuarios para probarlo

UVP → producto de valor único que quiero construir.
MVP → min. prod. viable

↓ un o
Sumatoria
de MVFs
(se vive con una sola
caract. son lo mismo)

Si no logro converger en algo. No hace falta coincidir en todo. Con las

piensan MMF
con (MRF caract min
de del release)

MMP: min. consistencia comercializable } o estás en condiciones de
MMP: producto min. comercializable } poner el producto o no independientemente de si cobras o no.
MVP: Quiero validar el producto pero una característica por vez (caract. min del producto)

→ lo que más se usa (lo min. con lo que el cliente quiere salir).

El foco no está puesto en el diseño o el código porque se supone que ya debe hacerse y con calidad.

MVP tiene que ver con la hipótesis → me voy a meter en un prod. que los usuarios
van a querer usar o no?

→ No necesitan código puede ser un video o Powerpoint.

audacia cero

→ riesgo. No tenemos muchas veces una segunda oportunidad.

Valor vs Desperdicio → Luego cosa que no genera valor es un desperdicio

Filosofía Lean → enfocada en la mejora continua de procesos
La calidad del producto es una determinante del producto →
que haga con ese proceso.

medio por el cual
hacemos las cosas

User: es un boletín para una conversación

2 ceremonias de SCRUM

Una para el producto (Screen review)
una para el proceso.

Si elim. el desperdicio mejora.

Filosofía Lean habla de 7 u 8 desperdicios.

(Time to Market) El 8 es el talento mal utilizado tanto desperdicio.
Tiempo de mercado: Ventana de oportunidad que tienes para salir con este punto. Que valora un producto con MVP, tener que salir rápido por la competencia. Ser rápido, aprender y otra vez.

Alinear el producto hacia algo que le de valor al cliente más rápido.
Nunca termina el aprendizaje sobre el prod, ni sus oport de mejorar - no termina el ciclo. Cada vez se puede hacer más cosas.

un prototipo podría ser un MVP, pero no se puede poner producción cuando lo vas a sacar a producción el soft funcional.

clave salir con algo rápido para validar
dilema: La quiebra de vuelta a cero

Cuando sales que no tienes nada que perder.

Dejar claras las hipótesis y muchas veces basadas en esta ~~idea~~ idea de la quiebra de vuelta a cero.

Salta de fe y tienen 2 hipótesis → De valor: tasa de retención

→ De creación: si tengo cada vez más usuarias.

No vamos a crear un prod que le sirva a todo el mundo tenemos que ele-
gir un nicho de mercado.

05/09

Recicla tus prendas

- Registrar venta
- Iniciar sesión X
- Filtrar prendas X
- Elegir prendas vendidas
- seleccionar prendas
- visualizar prendas
- Registrar cobro
- Notificar X

Hipótesis

Existe un número de vendedores aceptable que esté dispuesto a vender sus prendas usadas en nuestro sistema.

Corregido 6

Hipótesis: Hay vendedores dispuestos a poner en venta sus prendas que no utilizan a través de una empresa que se ocupa de todo el proceso de recepción, selección y comercialización de prendas.

Validar el numero de vendedores dispuestos a vender sus prendas a la empresa

Visualizar número de vendedores
Gestionar logística de envíos
Registrar venta de prendas por vendedor
Registrar ventas
Visualizar ventas en la página
Registrar precio por prendas
Registrar cobro

Alcance:

Desarrollar una aplicación web que permita registrar el catálogo de prendas para ser seleccionadas por la empresa, recibir las selecciones y digitalizar las prendas a vender y generar una propuesta de venta al vendedor.

US incluidas

- * - Registro usuario vendedor con mail.
- * - Cargar prendas para vender (formulario)
- * - Confirmar propuesta de venta
- * - Seleccionar prendas para publicar (análisis de selección)
- * - Recibir mail de propuesta de venta
- * - Recibir mail de prenda más vendida
- * - Completar ficha de prenda a publicar (análisis de publicación)

User de Tareas

- Completar ficha de prenda a publicar { Edición (para esa elegir la canónica) y justificar }
- Pagar al vendedor

No incluye

- Notificaciones
- filtros
- Funcionalidad de venta y logística
- Registro con Google
- Registro de USCs compradora,
- Funcionalidad de soporte
- Reportes.

Varios { Se suman
Proximos { Se leen
para no se { Se renuncian
este

08/09

Parcial 23 de Septiembre

Componentes de un proyecto de Software

El proyecto de software tiene ciertas características por las cuales es diferente, es complejo, por ejemplo:

Proceso definido: La retroalimentación viene más tarde, no podés modificar la experiencia tras la ejecución de un proyecto. El proceso lo define gente que no va a hacer el proyecto.

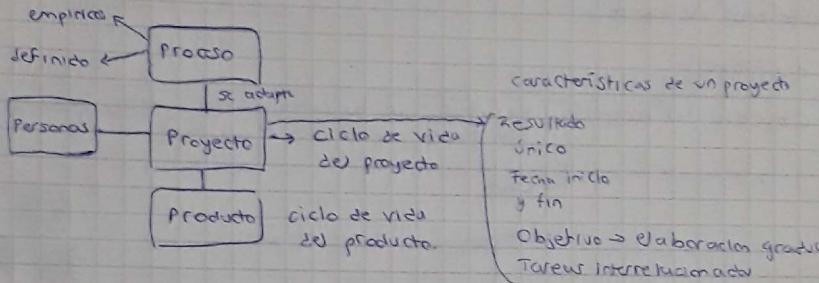
Proceso empírico: Responde a las respuestas en el equipo. Experiencia no es extrapolable a otros equipos, en las definiciones consideran que si plantean que si se cambia algún variable las resultados no van a ser los mismos (en empírico).

Proceso completo que tiene todos los capítulos que se pueden prever de antemano planificarse y acompañando a ello deben tener la posibilidad de adaptarse.

Si el proyecto identifica oportunidades de mejora para el proyecto va a tener en cuenta.
en las definidas.

En los empíricos hay def. de proceso en inicio del proyecto y si al final de la iteración
veremos de hacer cambios los hacemos.

SCRUM es un framework de trabajo no quiere ser un proceso definido o metodología.
se toman como base



Define el conj. de estados por los que pasa el proyecto desde que inicia hasta que decide terminarlo. Orden según el cuál tiene que hacer los cambios en un proceso.

Hay un montón de ciclos de vida pero se pueden agrupar en 3 tipos básicos

Morganatico
aceptar lo
que funciona

Tipo → Secuencial : cascada

Iterativo : iterativo increm.

Recurso : Espiral

Yo puedo elegir cuálq ciclo de vida y cuálq proceso. El PUD recomienda el iterativo increm.

Cuálq framework sigil barado el proceso empírico debe usar un ciclo iterativo por lo retroalimentación.

El ciclo de vida iterativo de increm de PUD y SCRUM no son iguales. Cambia el como
preparan llegar al objetivo

Iterativos → Duración fija (como max 1 mes si define en principio y no se cambia)
↓
SCRUM

→ Alcance fijo (se puede posponer la fecha)

↓
PUD

El cambio está en como se encara la resolución del problema.

El software se hace a través de proj. El soft se asimila con los conceptos que
tiene que tener un proyecto.

El soft es una actividad humana intensiva, el resultado de un proyecto es único y dife-
rente cada vez, va a tener variaciones. No se puede mantener linea de producción.

Por eso se hace a través de unidades de gestión que son los proyectos.

Un proyecto tiene una fecha inicio y fin definida.

Un objetivo que es de elaboración gradual

Tareas interrelacionadas

Empresarial del

El proceso se instancia en el proyecto



Como se obtiene hacia los cuales tecnia.

Un proy que se planifica a veces fracasa, uno que no, está destinado a fracasar.

El agilismo planifica más aún.

La primera que tenemos que definir para un proyecto es un nombre porque da pertenencia al equipo.

Plan del Proyecto

Lo objetivo del proyecto: Resultado que buscamos alcanzar. Neces. que voy a satisfacer. (Relac con el resultado único).

→ Claro

→ Alcanzable: Debe ser posible.



→ Alcance del proyecto: Todo el trabajo y solo el trabajo que tengo que hacer para alcanzar el objetivo.

Tanto el objetivo como el alcance

Deben de ser explícitos en un ~~artefacto~~ que se llama plan de proyecto o plan de desarrollo de Software.

En los procesos empíricos no hay una definición cerrada de los reg del producto, a lo mejor una visión. No tienes un lugar donde están definidos los carat del prod como es la ERS para los definidos.

El product backlog no es una def de requerimientos

Objetivo y alcance del producto están en la ERS.

Hay relación entre el alcance del proy y el del prod. pero no son la misma puedes tener uno sin tener el otro.

Definición del proceso y del ciclo de vida

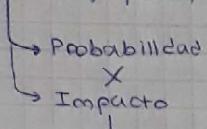
Estimaciones: Universo de incertidumbre (probabilidad). No hay certeza

Alexis
se estima
con un
orden y
hay qx re
petirlo

Recurso critico
(puede ser una licencia, el
mismo de fondo, un suger)

1) tamaño del producto
2) Esfuerzo (hs persona /meses) → asume que es una sola persona haciendo las cosas 1 por vez.
3) Tiempo /Calendario: analiza la gente que tengo, sus capacidades, los horas a hacer y su orden y establecimiento /los dependencias. Luego de peq análisis una fecha de cuando se podría terminar.

Riesgo: Probabilidad de ocurrencia de un evento que puede ocurrir en mi proyecto.
Riesgo prob 1 → Problema.



↓
Exposición: medida de comparación de los riesgos. valor de homogeneidad entre los riesgos

El mayor riesgo es no definir bien los requerimientos

También es un riesgo hacer malas estimaciones (ser muy optimistas).

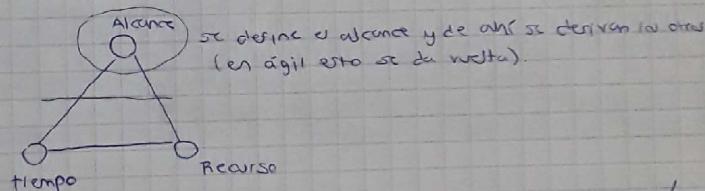
Necesito definir acciones de mitigación para bajar el nivel de exposición.
Ya sea hacer cosa para bajar la probabilidad o impacto.

No todos los riesgos tienen la misma ventaja temporal. Pueden ser riesgos en el momento y después no.

Métricas: Sirve para visualizar el estado del producto o el proyecto.
Tengo que elegirlas bien, deben ser útiles, calculables.
se las debe listar y explicar la forma de cálculo.
Las de ágil miden sistemas o no el producto.

Seguimiento: Todo lo que necesitas para hacer seguimiento del proyecto se define en el plan.

Trabaja este plan con la triple restricción



se define el alcance y de ahí se derivan los otros (en ágil esto se da vuelta).

Completar ficha de prenda a publicar

13/09

Como analista de publicación quiero llenar la ficha para una determinada prenda para poder publicarla en el ecommerce.

Enteros de Aceptación

Se debe poder identificar la prenda a través de QR.

Se debe cargar fotos pudiendo ser 5 como máximo.

Las fotos pueden pesar hasta 1 Mb.

Se debe cargar nombre de la prenda.

Se debe indicar un peso de la prenda en kg y debe ser mayor a cero.

Se puede editar el precio sugerido.

El precio sugerido se calcula en base a una lista de precios por categorías.

materias, estacionalidad y estilo de la prenda.

Se pueden llenar otras datos dinámicos.

Se debe poder visualizar la prenda.

paciente
común
número
de articulos
y QR por
categorías
y se
cambie.

Pruebas de Aceptación

Probar llenar la ficha con 3 fotos de menos de 1Mb c/u, cargando nombre, PESO, llenando el color y sin modificar el precio sugerido (pues).

Probar llenar la ficha modificando el precio sugerido (pues).

Probar llenar ficha sin cargar el nombre de la prenda (fallo).

Probar llenar una ficha sin escanear código QR (fallo).

Probar llenar una ficha con 7 fotos (fallo).

Probar llenar una ficha con fotos c/u de más de 1Mb (fallo).

Probar llenar una ficha sin peso (fallo).

Probar llenar una ficha con un peso de -5 (fallo).

Cuadrícula: Administración Categorías

Considerando esta como la user menos compleja ya que tiene un solo campo, no requiere demasiado esfuerzo su implementación y tiene un bajo nivel de error porque no hay prácticamente incertidumbre.

Teniendo en cuenta que a esta le vamos a dar un valor de 1, vamos a comparar contra esta la user de completar ficha de prenda o publicar.

La user de completar ficha de prenda o publicar la complejidad es un poco superior a la cuadrícula ya que requiere cargar fotos con sus validaciones y son varios campos con el cálculo del precio, además debe poder hacerse uso de QR.

El esfuerzo que va a ser necesario también va a ser mayor ya que son muchos campos y sus validaciones. Además puede que haya un poco de incertidumbre y dudas con respecto a la visualización del QR y al poder subir fotos.

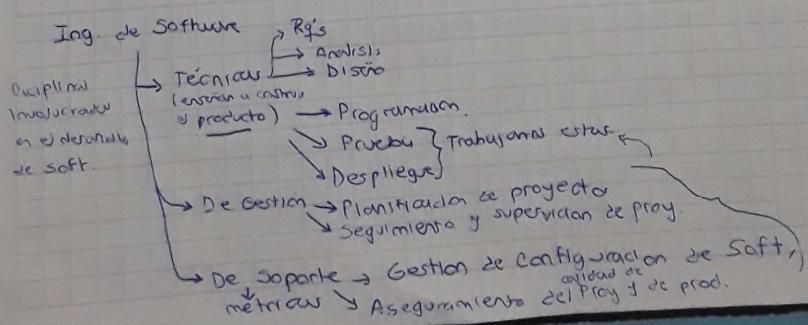
Por todo esto ya le dasca una complejidad de 5
(Aclarar si es baja - Medio - Alta)

TP N° 4 → evitable

Parcial 23/09

15/09

Software Configuration Management (SCM)



La gestión de config se hace en el contexto de proyecto.
En esto los items de config son los artefactos que salen de las tareas de un proyecto.

El proceso es algo teórico que te dice que hacer.

Software : Antes durante y después del proyecto se genera software (BD, diccionario de esp. de usuario, currículum de prueba).
El soft es conocimiento en dist. niveles de abstracción.
Es ubicuo → está en todas lados (no se lo puede localizar en un lugar)
Un plan de proyecto y un cronograma también es soft.
Cualquier artefacto que se genera como resultado de una tarea de proyecto, información → tiene contexto y relevancia.

Todo lo que es importante para mí y lo puedo guardar en un disco es un item de config (es un word.)

Caract fundamental del soft es que es complejo (y no tiene existencia física), es fácil de modificar (fácil alterarlo).

Lo que hace que el soft se degrada en su estructura son las combinas que se van haciendo en él.

SCM SURGIÓ en 1980 y es una disciplina transversal (tomas → estar involucradas desde el momento cero) es responsable de todo que tengas que una pieza de soft. Mientras el prod exista tienes que gestionar su config.

Da el cliente de que los items de config del proj tienen integridad y sobre eso podamos cumplir el aseguramiento de calidad (mientras se está haciendo). No vamos a evitarte el testing (los personas cometen errores).

Identificación de defectos cuya presencia se asume en el soft
La prueba se agrega a la disciplina de aseguramiento de calidad, pero la idea es la detección temprana.

Los problemas de comunicación con el cliente no hay herramientas que te lo solucionen.

Gestión de Configuración de Soft:

Disciplina ~~sistema~~ cuya propósito es mantener la integridad del producto de soft.

- en un momento dado se refiere a la versión, hay que identificarla.

- Controlar las combinaciones

- Mantener su integridad → tiene que cumplir ciertas cosas para mantener su integridad

Agregar que satisfaga las RNF y las funciones
rastreabilidad: establecer vínculos de trazabilidad. se debe establecer desde el momento cero.

trazabilidad para analizar el impacto.

vincular de info entre las vinculaciones de configuración.

Quiere resolver los problemas en el manejo de componentes (de su ppt).

De soporte porque ayuda "que lo que querés hacer salga mejor".

Item de config: Cualquier cosa que pueas guardar en un disco duro
debe tener una versión (número que me permite saber si me actualizo de un cambio)
Necesitamos que un componente que construyamos funcione en dist. o sea variante
(se arregla admira la forma de variante) con decisión del equipo.

Config del software: como instantánea a todos los items es config ^{con su versión} en un momento determinado

La gestión de conf busca mantener adm de la conf de ese soft a lo largo de su vida

Repositorio: Contenedor de items de conf y administra las herramientas de versionado.

Modelo de repositorio centralizado, bajo copia y modif y cuando está lista subo
la mayoría de las herramientas de gestión de config tiene de estos.
otros permiten partir..

Línea base: Es un item o conj de items de conf con su versión que están
en un estado de aprobación (su estado de terminación es tal que se puede
usar de referencia por ser estable). No nece. incluir código estos items.
se marca etiquetando todos los items que forman parte de esa línea base.

Son momentos referentes de evolución del producto
No es un item en una nuova que le hago a los items que quiera incluir para de
pues poder recuperarlos.

El item 1 de la gestión de config del proyecto sirve el plan de gestión de conf.
No se puede combinar sin el Comité de control de cambios.

De especificaciones: Tienen documentación

De productos: Ya contienen código que se va a liberar
no pasan por
un control de calidad

Ramas: las líneas base se ponen por el tronco, las ramas de versionado deben
valer al tronco. Si pre de autorizar el mezclado (merge de código). O pueden
eliminararse.

Hay 4 actividades básicas de la Gestión de Conf.

Identificación: definir repos, definir donde va cada item del repos,
se generan items con dist ciclos de vida, los que transcurren el proyecto,
los del proyecto mientras dura este y los de iteración menor.

Ej: DEAT-Arquitectura.docx → ej de item de config ^{cuyo} ciclo de vida dura más
que el proyecto.

Para los items que no son únicos se define un esquema de nombreado
DEAT-US «Fase Verbo».docx

Para un proj un ej: AD-PlanProyecto.docx
Importante identificar extensión de tipo.

Sig actividad es el control de cambios
se hace sobre la linea base, solo sobre esos items.
El comité de control de cambios definen si se realiza un cambio o no, evolucion.

En la gestión tradicional también se pide hacer cambios.

El control de estos permite organizarlos, ver si se pueden hacer
el comité lo conforman todos los que tienen algo que decir sobre ese cambio.
Gestion de conf, lider entre otros. Si el cambio es muy grande involucrar al cliente

Auditarios: Proceso de control que se realiza contra estandares. Buscar que se contraten con el plan de gestión de configuración para ver si hizo las cosas como dijo que iba a hacer. Si no hay plan no puedo auditar, se audita una linea base, quien va a auditar no debe tener relación con el equipo

De conf física: Intenta ver integración del repositorio. Se hace primero verificación de que los items existen y se les hace control, son transables.
Si la física no sale bien no se puede hacer la funcional. Busca revisar contra los requerimientos.

Informes de Estado: Poder sacar info del estado de situación de mi repositorio para saber para un cambio que lo hice y cuando. La mayoría se sacan de la herramienta

19/09

Gestión de Configuración

Tratar de ir trabajando con la última versión para hacer cambios chiquitos de poco impacto.

Si surge un error vuelve a la linea base

La linea base es una versión

Repositorio tiene estructura que me facilita encontrar los items de conf y usamos un esquema de nombrado para que sea fácil trabajar sobre eso.

Como nos sirve la estructura de nombrado

Primeras divisiones en Producto y Proyecto

Producto

- Código Fuente
- Arquitectura (siempre debes estar actualizada y debes tener la última versión definida)

Proyecto

- Product Backlog
 - ↳ User Stories (sin temporales)

- Métricas

- Sprint

- ↳ Sprint Backlog (que users se van a trabajar en ese sprint)

- ↳ Métricas

- Sprint N

- ↳ Código fuente

Items - Reglas de Nombre

Clasítico por prod, proyecto y sprint.

Sprint N - Backlog.jpg

Poner en referencias que es HU o NN o fraseVerbal...

HU - NN - FraseVerbal.docx

versión : conj de ítems de conf con su versión que en un momento querí entrega para desparr su release. Para tener trazabilidad. Debe tener código, haber sido testeado.

una linea base puede no ser una versión.

Hay versiones que no llegan a release porque tienen un error y nunca se ponen en producción.

se puede tener muchas versiones internas antes de tener un release (entregable).

Variante: Hacer lo mismo en diferentes plataformas (es web o mobile).

branch: No querés afectar config del main.

Cuando necesa clarificar datos y no los tener disp hay incertidumbre.

Efuerzo: cont horas ideales para completar user. siendo una persona que hace una cosa por vez.

El tiempo no es una variable cuando estimas con story point porque es fijo.

No necesa todo MVP validado se vuelve en un comendable (MMP). A veces no estan llenos.

Scrum

Es un framework de trabajo para gestionar proyectos
Es un marco de trabajo liviano para el desarrollo de productos complejos
Ayuda a la gestión ágil de proyectos espec cuando son complejos

Roles

- Product owner → No técnico
- Scrum master
- Developers (más genérico que los desarrolladores)
↳ min 5 integrantes al equipo

siguen los pilares del empirismo

Los valores son: caraque, foco, compromiso, respeto y ..

Deben tenerlos quienes pertenece al equipo.

Transparencia ser sinceros, decir si necesa ayuda

↳ brinde de las otras 2

4 ceremonias de Scrum

reuniones

eventos

Una adicional que es continua (en el sentido que se convoca cuando se necesita)

Refinamiento

El scrum master es un developer y se lo requiere en todas las ceremonias. No requiere ser el referimiento si no hace nada técnica.

Cada artefacto con su compromiso

Compromiso → lo que uso para contactar si se cumple o no

Sprint backlog → su compromiso es el sprint goal

El siguiente artefacto es el incremento y su compromiso es el DoD.

Prim. ceremonia

Sprint Planning → obj de definir sprint backlog

Daily Scrum → No es obligatorio que este el PO.

(15 min max) → Averiguar sincronizarnos, que hicieron en la anterior reunión, etc
y contar si hay algún problema (transparencia, respeto)

Presenta los DoD (100%) terminados. Si el product owner les usa para medir la velocidad. Si me los redacta los vuelvo a poner el product backlog
La velocidad no se estima se calcula después de la Review.

Retrospective: Reunión de inspección y adaptación sobre el proceso, mejorar prácticas que debieron hacer o no deberíamos haber hecho..

Algunas usan juegos o dinámicas. Aspectos positivos o neg. del proceso (no de la gente)

Si solo sabemos scrum no podemos hacer soft.

Asumir equipo ya formado (capacitado)

Líder de Proy
↓
Jefe/guru superior del equipo
↓
Tiene tu responsabilidad
↓
Sist push

Equipo autogestionado / Sist pull, el scrum master es un líder de servicio.
El scrum master puede ser rotativo. Ayuda a resolver impedimentos, charla más con el product owner, reserva la sala para una reunión.

→ menos Jefes

El calendario de todas las ceremonias es el Sprint (tiene duración máx de 30 días)

Time box → dejar fijo el tiempo

Las ceremonias pueden tener un determinado tiempo como máx.

En la planificación de un sprint aparecen todos los participantes del equipo.
La velocidad no es una garantía. Deberá llegar un punto en que se estabilice y
llegar a un desarrollo sostenible (si vengo queriendo 10 puntos de historia) por ej.
Capacidad → se estima y es la medida del compromiso de trabajo que puede
asumir un equipo. Podría ser un Story point.
Las más inmaduras en horas ideales

Dién para otras actas de scrum. Se tiene a la persona pero no se puede asignar para actos.